Preparation:

[Python / Numpy Tutorial](http://cs231n.github.io/python-numpy-tutorial/)

[IPython Notebook Tutorial](http://cs231n.github.io/ipython-tutorial/)

Setup Instructions

Setup

You can work on the assignment in one of two ways: locally on your own machine, or on a virtual machine on Google Cloud.

Working locally

**Installing Anaconda:** If you decide to work locally, we recommend using the free [Anaconda Python distribution](https://www.anaconda.com/download/), which provides an easy way for you to handle package dependencies. Please be sure to download the Python 3 version, which currently installs Python 3.7. We are no longer supporting Python 2.

**Anaconda Virtual environment:** Once you have Anaconda installed, it makes sense to create a virtual environment for the course. If you choose not to use a virtual environment, it is up to you to make sure that all dependencies for the code are installed globally on your machine. To set up a virtual environment, run (in a terminal)

conda create -n cs231n python=3.7 ~~anaconda~~

上述命令末尾不要加anaconda,否则会自动安装很多常用的包，这些包的版本不符合作业要求，不加anaconda，默认情况下仅仅安装一些必须的包

to create an environment called cs231n.

Then, to activate and enter the environment, run

source activate cs231n

To exit, you can simply close the window, or run

source deactivate cs231n

Note that every time you want to work on the assignment, you should run source activate cs231n (change to the name of your virtual env).

You may refer to [this page](https://conda.io/docs/user-guide/tasks/manage-environments.html) for more detailed instructions on managing virtual environments with Anaconda.

Assignment 1:

In this assignment you will practice putting together a simple image classification pipeline, based on the k-Nearest Neighbor or the SVM/Softmax classifier. The goals of this assignment are as follows:

* understand the basic **Image Classification pipeline** and the data-driven approach (train/predict stages)
* understand the train/val/test **splits** and the use of validation data for **hyperparameter tuning**.
* develop proficiency in writing efficient **vectorized** code with numpy
* implement and apply a k-Nearest Neighbor (**kNN**) classifier
* implement and apply a Multiclass Support Vector Machine (**SVM**) classifier
* implement and apply a **Softmax** classifier
* implement and apply a **Two layer neural network** classifier
* understand the differences and tradeoffs between these classifiers
* get a basic understanding of performance improvements from using **higher-level representations** than raw pixels (e.g. color histograms, Histogram of Gradient (HOG) features)

## Setup

Get the code as a zip file [here](http://cs231n.github.io/assignments/2019/spring1819_assignment1.zip).

### Download data:

Once you have the starter code (regardless of which method you choose above), you will need to download the CIFAR-10 dataset. Run the following from the assignment1 directory:

cd cs231n/datasets

./get\_datasets.sh

### Start IPython:

After you have the CIFAR-10 data, you should start the IPython notebook server from the assignment1 directory, with the jupyter notebook command.

### Some Notes

**NOTE 1:** There are # \*\*\*\*\*START OF YOUR CODE/# \*\*\*\*\*END OF YOUR CODE tags denoting the start and end of code sections you should fill out. Take care to not delete or modify these tags, or your assignment may not be properly graded.

**NOTE 2:** The submission process this year has **2 steps**, requiring you to 1. run a submission script and 2. download/upload an auto-generated pdf (details below.) We suggest **making a test submission early on** to make sure you are able to successfully submit your assignment on time (a maximum of 10 submissions can be made.)

**NOTE 3:** This year, the assignment1 code has been tested to be compatible with python version 3.7 (it may work with other versions of 3.x, but we won’t be officially supporting them). You will need to make sure that during your virtual environment setup that the correct version of python is used. You can confirm your python version by (1) activating your virtualenv and (2) running which python.

**NOTE 4:** If you are working in a virtual environment on OSX, you may potentially encounter errors with matplotlib due to the [issues described here](http://matplotlib.org/faq/virtualenv_faq.html). In our testing, it seems that this issue is no longer present with the most recent version of matplotlib, but if you do end up running into this issue you may have to use the start\_ipython\_osx.shscript from the assignment1 directory (instead of jupyter notebook above) to launch your IPython notebook server. Note that you may have to modify some variables within the script to match your version of python/installation directory. The script assumes that your virtual environment is named .env.

### Q1: k-Nearest Neighbor classifier (20 points)

The IPython Notebook **knn.ipynb** will walk you through implementing the kNN classifier.

### Q2: Training a Support Vector Machine (25 points)

The IPython Notebook **svm.ipynb** will walk you through implementing the SVM classifier.

### Q3: Implement a Softmax classifier (20 points)

The IPython Notebook **softmax.ipynb** will walk you through implementing the Softmax classifier.

### Q4: Two-Layer Neural Network (25 points)

The IPython Notebook **two\_layer\_net.ipynb** will walk you through the implementation of a two-layer neural network classifier.

### Q5: Higher Level Representations: Image Features (10 points)

The IPython Notebook **features.ipynb** will walk you through this exercise, in which you will examine the improvements gained by using higher-level representations as opposed to using raw pixel values.